Learning Diagnosis Based on Evolving Fuzzy Finite State Automaton

Moussa Traore[1], Eric Châtelet[2], Eddie Soulier[3], and Hossam A. Gabbar[4]

[1,2,3] *University of Technology of Troyes (UTT), Troyes, BP 2060, 10010, France*
moussa_amadou.traore@utt.fr
eric.chatelet@utt.fr
eddie.soulier@utt.fr

[4] *Faculty of Energy Systems and Nuclear Science, University of Ontario Institute of Technology,*
2000 Simcoe St. North, Oshawa, Ontario, Canada L1H7K4
Hossam.Gaber@uoit.ca

ABSTRACT

Nowadays, determining faults (or critical situations) in non-stationary environment is a challenging task in complex systems such as Nuclear center, or multi-collaboration such as crisis management. A discrete event system or a fuzzy discrete event system approach with a fuzzy role-base may resolve the ambiguity in a fault diagnosis problem especially in the case of multiple faults (or multiple critical situations). The main advantage of fuzzy finite state automaton is that their fuzziness allows them to handle imprecise and uncertain data, which is inherent to real-world phenomena, in the form of fuzzy states and transitions. Thus, most of approaches proposed for fault diagnosis of discrete event systems require a complete and accurate model of the system to be diagnosed. However, in non-stationary environment it is hard or impossible to obtain the complete model of the system. The focus of this work is to propose an evolving fuzzy discrete event system whose an activate degree is associated to each active state and to develop a fuzzy learning diagnosis for incomplete model. Our approach use the fuzzy set of output events of the model as input events of the diagnoser and the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment.

1. INTRODUCTION

A great number of systems or situations can be naturally viewed as discrete event systems. A discrete event system is a dynamic system whose the behavior is governed by occurrence of physical events that cause abrupt changes in the state of the system (Liu & Qiu, 2009a; Cassandras & Lafortune, 1999; Moamar & Billaudel, 2012; Traore, Moamar, & Billaudel, 2013). Discrete event system theory, particularly on modeling and diagnosis, has been successful employed in many areas such as concurrent monitoring and control of complex system (Cao & Ying, 2005). Usually, a discrete event system is modeled by Automaton (Dzelme-Berzina, 2009; Mukherjee & Ray, 2014) or Petri Net (Patela & Joshi, 2013). Automaton (or more precisely a finite state automaton) are the prime example of general computational systems over discrete spaces and have a long history both in theory and application (Thomas, 1990; Moghari, Zahedi, & Ameri, 2011). A finite state automaton is an appropriate tool for modeling systems and applications which can be realized as finite set of states and transition between them depending on some input strings (Doostfatemeh & Kremer, 2004). And, the behavior of discrete event system modeled by an automaton is described by the language generated by the automaton.

Discrete event systems are divided into two categories: crisp discrete event system and fuzzy discrete event system. A crisp discrete event system is usually described by a deterministic automaton (Luo, Li, Sun, & Liu, 2012) and fuzzy state is the extension of crisp discrete event system by proposing fuzzy state and every state transition is associated with a possibility degree, called in the following membership value. Thus, the membership value can be defined as the possibility of the transition from current (active) state to next state. The main advantage of fuzzy finite state automaton is that their fuzziness allows them to handle imprecise and uncertain data, which is inherent to real-world phenomena, in the form of fuzzy states and transitions. In literature, many application of fuzzy discrete event system had been proposed (Gerasimos, 2009; Luo et al., 2012; Sardouk, Mansouri, Merghem-Boulahia, & Gaiti, 2013). Thus, one of the interesting characteristics of fuzzy automaton is the possibility of several transitions from different current fuzzy states lead to the same next fuzzy state simultaneously, and also the possibility of several transitions from one current fuzzy state lead to the different next fuzzy states simultaneously and consequently several output label can be activated at the same time (Doostfatemeh & Kremer, 2005). For this reason, fuzzy discrete event is very adapted to resolve the ambiguity in a fault diagnosis problem especially in the case of multiple faults. In this paper, these output events constituted of a fuzzy set are applied as input event for our diagnoser. Most of applications, the output should be crisp. Therefore, the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment. Thus, the outputs are assumed to be observable.

The diagnosis of discrete event systems is a research area that has received a lot of attention in the last years and has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems (Cabasino & Alessandro Giua, 2010) or complex situation (like crisis situation) (Traore et al., 2013). Hence, the use of finite state automaton in fault diagnosis tasks has gained particular attention in the case of discrete event dynamic systems (Gerasimos, 2009). Although, most of approaches proposed in literature for fault diagnosis of discrete event systems require a complete and accurate model of the system to be diagnosed. However, the discrete event model may have arisen from abstraction and simplification of a continuous time system or through model building from input-output data. As such, it may not capture the dynamic behavior of the system completely. Therefore, in this paper, we attempt to develop a diagnosis approach based on fuzzy automaton for incomplete model in non-stationary environment. For most of real-world applications operate in non-stationary environment.

The diagnosis approach proposed in our paper is different from the approach proposed in (Kwong & Yonge-Mallo, 2011). In our paper, the diagnoser is a finite-state Automaton which takes fuzzy output sequence of the system as its input. Here, the learning diagnoser is constructed off-line and the diagnosis is performed on-line using input and output data generated by system's model. The on-line diagnosis system allows to build an evolving fuzzy finite state system by updating the set of states and/or the set of input symbols. The new states and/or transitions detected by the diagnoser is validated by an expert of the system or situation.

The potential application of learning diagnosis based on fuzzy finite state automaton is in solving the ambiguity in a fault diagnosis problem especially in the case of multiple faults.

This paper is organize as follows. In section 2 , we present the required background of crisp discrete event system. We describe the general definition for fuzzy discrete event system in section 3 . The standard diagnoser is presented in section 4. The algorithm of the learning diagnosis based on evolving fuzzy finite state automaton is proposed in section 5. Learning diagnoser application to crisis management is presented in section 6.

2. CRISP DISCRETE EVENT SYSTEM

A crisp discrete event system is usually described by a deterministic automaton $G = \{X, \Sigma, \varphi, Y, x_0, F\}$, where

- X is the set of states
  $X = \{x_0, x_1, \cdots, x_{n-1}, x_n\}$,
- Σ is set of input symbols,
  $\Sigma = \{a_0, a_1, \cdots, a_{m-1}, a_m\}$,
- $\varphi : X \times \Sigma \to$ X is the transition function,
- Y is the set non-empty finite set of output,
  $Y = \{y_0, y_1, \cdots, y_{l-1}, y_l\}$,
- $x_0 \in X$ is the start state and
- $F \subseteq X$ is the (possibly empty) set of accepting or terminal states,

The event set Σ includes the set of failure events (or critical events) Σ_f (Kwong & Yonge-Mallo, 2011). In addition to the normal situation (mode) N, there are p critical situation (or failure mode) $F_1, \cdots, F_p$ that describe the evolution of the condition's system. We denote the condition set of the situation by
$\lambda = \left\{N, F_1, \cdots, F_p\right\}$, in this case, the state set partitioned into

$$X = X_N \cup X_{F_1} \cup \cdots \cup X_{F_p}.$$

In (Traore et al., 2013), we proposed the extension of the transition function φ represented as: $\varphi : X \times \Sigma \to X \times Y$.
Let φ_1 and φ_2 be the two projection of φ such as φ_1 gives the state reached from a state $x_i \in X$ and a given input $a_k \in \Sigma$ and φ_2 defines the output sequence from state x_i and input a_k. The expression of φ_1 and φ_2 are given by

$$\varphi_1(x_i, a_k) = \left\{x_j \mid \exists\, y_j \text{ such that } (x_j, y_j) \in \varphi(x_i, a_k)\right\},$$
$$\varphi_2(x_i, a_k) = \left\{y_j \mid \exists\, x_j \text{ such that } (x_j, y_j) \in \varphi(x_i, a_k)\right\},$$

where x_i, $x_j \in X$ and $a_k \in \Sigma$ and $y_j \in Y$. The new definition of φ is:

$$\varphi(x_i, a_k) = (\varphi_1(x_i, a_k), \varphi_2(x_i, a_k)).$$

These two projection may be extended to take input sequence, for example: $x_j \in \varphi_1(x_i, \sigma_i \in \Sigma^*)$ and/or output sequence for example: $\sigma_y \in \varphi_2(x_i, \sigma_i \in \Sigma^*)$, where $\sigma_i = a_1 a_2 \cdots a_l$ and $\sigma_y = y_0 y_1 \cdots y_n$. Σ^* is a set of all strings formed by events in Σ, example $a_k \in \Sigma$, then, $a_1 a_2 \cdots a_k \in \Sigma^*$.

The behavior of G is described by the language generated by G denoted as $\mathscr{L}(G)$ or simply by $\mathscr{L}$ (Liu & Qiu, 2009b).

3. FUZZY DISCRETE EVENT SYSTEM

Fuzzy discrete event systems as a generalization of (crisp) discrete event systems have been introduced in order that it is possible to effectively represent uncertainty, imprecision, and vagueness arising from the dynamic of systems. A fuzzy discrete event system has been modelled by a fuzzy automaton; its behavior is described in terms of the fuzzy language generated by the automaton (Cao & Ying, 2006).

A Fuzzy Finite Automaton (FFA) is a 6-tuple

$$\widetilde{G} = \{X, \Sigma, \delta, Y, \widetilde{x}_0, F\}.$$

i The fuzzy subset $\delta : X \times \Sigma \times X \rightarrow [0\ 1]$ is a function, called the fuzzy transition function. A transition from state x_i (current state) to x_j (next state) upon a_k with the weight ω_{ij} is denoted as: $\delta(x_i, a_k, x_j) = \omega_{ij}$,

ii $\widetilde{x}_0 \in X$ is the set of initial states.

One of the interesting characteristics of FFA is the possibility of several transitions from different current (or active) states lead to the same next state simultaneously (*see* Figure 1.(a)). Thus, the possibility of several transitions from one current states lead to the different next states simultaneously as shown in Figure 1.(b), and consequently several output label can be activated at the same time (Doostfatemeh & Kremer, 2005). It is possible to have more than one start state with FFA.

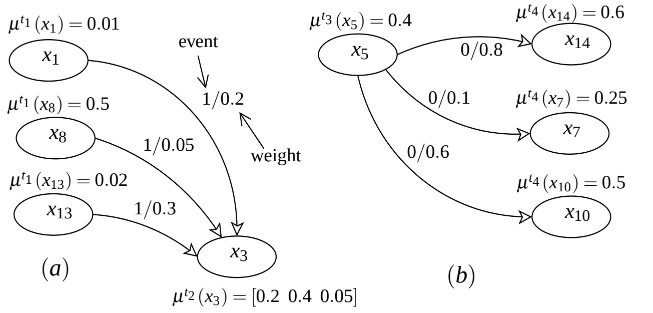

Figure 1. A example of FFA.

when an input a_k occurs at time t, all active state at this time, are those states to which there is at least one transition on the input event a_k. Then, the fuzzy set of all active state at time t is called active state set at time t. A active state set denoted X_{act} is consisted of state and their $mv's$. The definition of X_{act} is given by:

$$X_{act}(t) = \left\{ \left(x_j, \mu^t(x_j)\right) | \exists (x_i \in X_{pred}(x_j), a_k \in \Sigma) \wedge x_j \in X_{succ}(x_i, a_k) \right\},$$
$$X_{pred}(x_j) = X_{pred}(x_j, t) \text{ and},$$
$$X_{pred}(x_j, t) = \left\{ x_i \mid \exists\, a'_k \text{ s.t } x_j \in \varphi_1(x_i, a'_k) \ \wedge \ x_j \in X_{act}(t) \right\},$$
$$X_{succ}(x_i, a_k) = \left\{ x_j \mid x_j \in \varphi_1(x_i, a_k) \right\},$$
$$\delta(x_i, a_k, x_j) = \omega_{ij},$$

For example in Figure 1.(a)
$\varphi_1(x_1, 1) = \varphi_1(x_8, 1) = \varphi_1(x_{13}, 1) = x_3.$

where x_i is the state at time $t-1$, $\mu^t(x_j)$ is the membership of state x_j at time t, $X_{pred}(x_j, t)$ is all predecessors set of active state x_j and $X_{succ}(x_j, a_k)$ is all successors set of the state x_j on input symbol a_k. The successor $X_{succ}(x_j, a_k)$ is the set of all x_j which will be reached via transition function $\delta(x_j, a_k)$. In the following, all successors set of x_j is denoted by $X_{succ}(x_j, \overset{all}{\rightarrow})$, when the next state depend to the occurrence of different events.

We use the same notation for the active state, when the upon entrance is a string Γ. The active state set of the string Γ is given by:

$$X_{act}(\Gamma) = X_{act}(t_0 + |\Gamma|),$$

where $|\Gamma|$ represent the length of Γ.

Definition 1 *A fuzzy set Δ_X defined on a set X (discrete or continuous), is a function mapping each element of X to a unique element of the interval $[0\,1]$, $\Delta_X : X \rightarrow [0\,1]$. The membership value (mv) of the state $x_i \in X$ at time t is denoted as $\mu^t(x_i)$.*

For example in Figure 1.(a), at time time t_1, the active state is $X_{act}(t_1) = \{x_1,\ x_8,\ x_{13}\}$ and $X_{succ}(x_1, 1) = \{x_3\}$, $X_{succ}(x_8, 1) = \{x_3\}$ and $X_{succ}(x_{13}, 1) = \{x_3\}$, and at time time t_2, the active state is $X_{act}(t_2) = \{x_3\}$ and $X_{pred}(x_3, t_2) = \{x_1,\ x_8,\ x_{13}\}$, that mean the state x_3 is forced to take several different *mv* at this time. Hence, x_3 is a state with multi-membership, that we will call in the following multi-membership state.

In Figure 1.(b), each *mv* $\mu^{t+1}(x_j)$ of the state x_j at time $t+1$ is computed by using the function Ψ_1, named augmentation transition function. The function Ψ_1 should satisfy the two following axioms.

1. $0 \leq \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right) \leq 1$,
2. $\Psi_1(0,0) = 0$ and $\Psi_1(1,1) = 1$.

To compute $\mu^{t+1}(x_j)$, the function Ψ_1 use two parameters: $\mu^t(x_i)$ at time t and the weight ω_{ij} of the transition.

same example of Ψ_1 are:

- Arithmetic Mean

$$\begin{aligned} -\mu^{t+1}(x_j) &= \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right), \\ &= Mean(\mu^t(x_i), \omega_{ij}), \\ &= \frac{\mu^t(x_i) + \omega_{ij}}{2}, \end{aligned}$$

- Geometric Mean

$$\begin{aligned} -\mu^{t+1}(x_j) &= \Psi_1\left(\mu^t(x_i), \delta(x_i, a_k, x_j)\right), \\ &= GMean(\mu^t(x_i), \omega_{ij}), \\ &= \sqrt{\mu^t(x_i) \times \omega_{ij}}, \end{aligned}$$

where $\mu^t(x_i)$ is the *mv* of the corresponding predecessor of x_j and $\delta(x_i, a_k, x_j) = \omega_{ij}$.

The *mv* of each active state is used as the level of activation of each active state and the active state can be multi-membership state. However, in this paper, we need a single value for each active state. For this reason, the function Ψ_2 is introduced to compute the single *mv* corresponding to the state that was forced to take several *mv* by these predecessors. The single membeship value $\mu^{t+1}(x_j)$ of each multi-membership state given by:

$$-\mu^{t+1}(x_j) = \mathop{\Psi_2}\limits_{i=1}^{m}[\Psi_1(\mu^t(x_i),\omega_{ij})],$$

where *m* is the number of simultaneous transitions from states x_i to state x_j prior to time $t+1$.

The function Ψ_2 should satisfy the minimum requirements following axioms:

1. $0 \leq \mathop{\Psi_2}\limits_{i=1}^{m}[\Psi_1(\mu^t(x_i),\omega_{ij})] \leq 1,$

2. $\Psi(\phi) = 0,$

3. $\mathop{\Psi_2}\limits_{i=1}^{m}[\Psi_1(\mu^t(x_i),\omega_{ij})] = \nu$, if $\forall(\Psi_1(\mu^t(x_i),\omega_{ij}) = \nu),$

same example of Ψ_2 are:

- Maximum multi-membership resolution

$$-\ \mu^{t+1}(x_j) = \mathop{Max}\limits_{i=1\ to\ m}\left[\Psi_1(\mu^t(x_i),\omega_{ij})\right],$$

- Arithmetic mean multi-membership resolution

$$-\ \mu^{t+1}(x_j) = \frac{\left[\sum\limits_{i=1}^{m}\Psi_1(\mu^t(x_i),\omega_{ij})\right]}{m},$$

4. CASE STUDY

Consider the *FFA* in Figure 4 with several transition overlaps and several output labels. It is specified as:

$$\widetilde{G} = (X,\Sigma,\delta,Y,\widetilde{x}_0,F),$$

The dashed line in Figure 4, between states 12 and 13 represents a failure event or critical event. The occurrence of event $''f''$ bring the system in failure (or critical) mode corresponding to state x_{13}.

For instance, during the crisis management, the procedures designed by one or more organizations for the crisis situations can be applied, or partially applied or no applicable (no suitable) for the current situation. This latter case can be modeled by the state x_{13} in Figure 4 and for the reconfiguration, the model of crisis must be evolving and accepting missing information, whose the advantage to develop an evolving fuzzy finite state automaton for crisis management.

In this example

$$\begin{aligned} X &= \{x_0,x_1,\cdots,x_{13}\},\ \text{the set of states},\\ \Sigma &= \{a,b,c,d,e\},\ \text{set of input symbols},\\ Y &= \{\theta,\alpha,\beta,\gamma,\mu,\rho,\kappa,\xi,\eta\},\ \text{set of output},\\ \widetilde{x}_0 &= \{x_0,\mu^{t_0}(x_0)\},\ \text{fuzzy subset initial state},\\ \Delta_X &= \{0.04,0.09,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\}, \end{aligned}$$

$$\lambda(x_i) = \begin{cases} F_1, & \text{if}\quad \text{i=13}\ ,\\ N, & otherwise \end{cases}$$

we suppose, $\mu^{t_0}(x_0) = 1$ at the beginning and $\widetilde{x}_0 = \{(x_0,1)\}$ and all the other *mv* are computed by using the function Ψ_2 and/or Ψ_1.

Assuming that $\widetilde{G}$ starts operating at time t_0 and the next three input are **"a, e, d"** respectively (one at a time), active states and their $mv's$ at each time step are as follows.

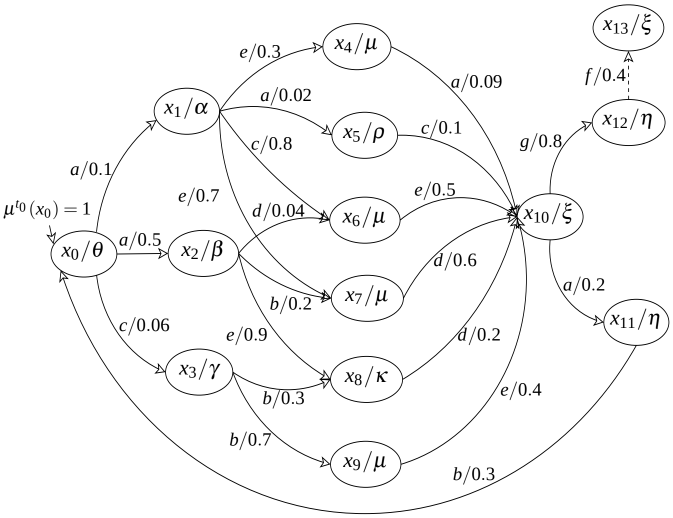

Figure 2. Fuzzy discrete event system model.

- **at time** t_0
$X_{act}(t_0) = \{(x_0,\mu^{t_0}(x_0)\}$ with $\mu^{t_0}(x_0) = 1,$

$$\begin{cases} X_{succ}(x_0,a_k) = \begin{cases} \{x_1,x_2\} & \text{if}\quad a_k = a,\\ \{x_3\} & \text{if}\quad a_k = c, \end{cases}\\ X_{succ}(x_0,\overset{all}{\rightarrow}) = \{x_1,x_2,x_3\}. \end{cases}$$

$X_{succ}(x_j,\overset{all}{\rightarrow})$ is the set of all (possible) successors of state x_j,

- **at time** t_1**, input is "a"**
$X_{act}(t_1) = \{(x_1,\mu^{t_1}(x_1)),(x_2,\mu^{t_1}(x_2))\},$

and

$$\left.\begin{aligned} X_{pred}(x_1,t_1) &= \\ X_{pred}(x_2,t_1) &= \end{aligned}\right\}\ \{x_0\}$$

and $|X_{pred}(x_1,t_1)|$ is the number of predecessors of state x_1, and

$$|X_{pred}(x_1,t_1)| = |X_{pred}(x_2,t_1)| = 1,$$

and when

$$|X_{pred}(x_j,t)| \leq 1,$$

the state x_j have a single *mv* and Ψ_1 is used to compute $\mu^t(x_j)$ for the state x_j, othewise the function Ψ_2 is used.

The *mv* of x_1 and x_2 is computed by:
$\mu^{t_1}(x_1) = \Psi_1(\mu^{t_0}(x_0), \delta(x_0,a,x_1) = \Psi_1(1,0.1),$
$\mu^{t_1}(x_2) = \Psi_1(\mu^{t_0}(x_0), \delta(x_0,a,x_2) = \Psi_1(1,0.5),$

and

$X_{succ}(x_1, \overset{all}{\rightarrow}) = \{x_4,x_5,x_6,x_7\},$
$X_{succ}(x_2, \overset{all}{\rightarrow}) = \{x_6,x_7,x_8\},$

- **at time t_2, input is ”e”**
  $X_{act}(t_2) = \{(x_4,\mu^{t_2}(x_4)),(x_7,\mu^{t_2}(x_7)),(x_8,\mu^{t_2}(x_8))\},$

  and

  $\mu^{t_2}(x_4) = \Psi_1(\mu^{t_1}(x_1), \delta(x_1,e,x_4),$
  $\mu^{t_2}(x_7) = \Psi_1(\mu^{t_1}(x_1), \delta(x_1,e,x_7),$
  $\mu^{t_2}(x_8) = \Psi_1(\mu^{t_1}(x_2), \delta(x_2,e,x_8),$

  and

  $X_{succ}(x_4,a) = X_{succ}(x_7,d) = X_{succ}(x_8,d) = \{x_{10}\},$

  and

  $X_{pred}(x_4,t_2) = X_{pred}(x_7,t_2) = \{x_1\},$
  $X_{pred}(x_8,t_2) = \{x_2\},$
  $|X_{pred}(x_4,t_2)| = |X_{pred}(x_7,t_2)| = 1$ and
  $|X_{pred}(x_8,t_2)| = 1,$
- **at time t_3, input is ”d”**
  $X_{act}(t_3) = \{x_{10},\mu^{t_3}(x_{10})\},$

  and

  $X_{pred}(x_{10},t_3) = \{x_4,x_7,x_8\}$, & $|X_{pred}(x_{10},t_3)| \geq 1$, hence, the state x_{10} is forced to take several different *mv*, then Ψ_2 is used to compute $\mu^{t_3}(x_{10})$.

$$\begin{cases} \mu_1(t_3) = \Psi_1(\mu^{t_2}(x_4), \delta(x_4,d,x_{10})), \\ \mu_2(t_3) = \Psi_1(\mu^{t_2}(x_7), \delta(x_7,d,x_{10})), \\ \mu_3(t_3) = \Psi_1(\mu^{t_2}(x_8), \delta(x_8,d,x_{10})), \\ \mu^{t_3}(x_{10}) = \Psi_2\left[\mu_1(t_3),\mu_2(t_3),\mu_3(t_3)\right], \end{cases}$$

to compute $\mu^{t_3}(x_{10})$, we can use Maximum multi-membership resolution given by relation (3) or Arithmetic mean multi-membership resolution defined by relation (3).

The fuzzy set of all active output, *i.e.*, output labels together with their $mv's$, at time t denoted as $Y_{act}(t)$, is called the active output set at time t, given by:

$$Y_{act}(t) = \left\{(y_l, \tau^t(y_l)\right\} \text{ and } Y_{act}(\Gamma) = Y_{act}(t_0 + |\Gamma|),$$

where $\tau^t(y_l)$ is the grade membership of the output y_l at time t. In this paper, y_l can be a state with multi-membership. For example,

- at time t_1

$$\begin{aligned} Y_{act}(t_1) &= \left\{(\alpha, \tau^{t_1}(\alpha)),(\beta, \tau^{t_1}(\beta))\right\}, \\ &= \left\{(\alpha, \mu^{t_1}(x_1)),(\beta, \mu^{t_1}(x_2))\right\}, \end{aligned}$$

- at time t_2, the active state x_4 and x_7 generate the same output label μ, *i.e.*, see Figure 4

$$\begin{aligned} Y_{act}(t_2) &= \left\{(\mu, \tau^{t_2}(\mu),(\kappa, \tau^{t_2}(\kappa))\right\}, \\ &= \left\{(\mu, [\mu^{t_2}(x_4) \;\; \mu^{t_2}(x_7)]),(\kappa, \mu^{t_2}(x_8))\right\}, \end{aligned}$$

most of applications, the output should be crisp. Therefore, the output of a fuzzy system should be defuzzified in an appropriate way to be usable by the environment and the outputs are assumed to be observable.

A diagnoser must be able to detect and isolates faults and failures (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). In this paper, the diagnoser $D_{\widetilde{G}}$ is a finite-state Automaton which takes the fuzzy output sequence of the system, *i.e.*, $\{(y_1, \tau^{t_1}(y_1), \cdots, (y_k, \tau^{t_k}(y_k))\}$ as its input, and based on this sequence calculates a set $z_k \in 2^X - \{\emptyset\}$ to which $x_i \in X$ must belong a time that pair $(y_k, \tau^{t_k}(y_k))$ was generated. The diagnoser $D_{\widetilde{G}}$ is given by:

$$D_{\widetilde{G}} = (Z, Y, \zeta, \lambda, z_0, \Omega),$$

with

- Z is the set of standard diagnoser state,
- Y is the set of standard diagnoser input,
  we recall, Y is the output of model $\widetilde{G}$,
- λ is the set of standard dianoser output,
- $\zeta : Z \times Y \times \rightarrow Z \times \lambda$ is the standard diagnoser state transition function,
- z_0 is the start state set of the standard diagnoser,
- $\Omega \in Z$ is the (non-empty) set of terminal states

Let ζ_1 and ζ_2 be the two projections of ζ of $D_{\widetilde{G}}$, with ζ_1 and ζ_2 are given by

$$\begin{cases} \zeta_1(z_k,y_{k+1}) = \{z_{k+1} \mid \exists\, \lambda_i \wedge (z_{k+1},\lambda_i) \in \zeta(z_k,y_{k+1})\}, \\ \zeta_2(z_k,y_{k+1}) = \{\lambda_i \mid \exists\, z_{k+1} \wedge (z_{k+1},\lambda_i) \in \zeta(z_k,y_{k+1})\}, \\ \zeta(z_k,y_{k+1}) = (\zeta_1(z_k,y_{k+1}), \zeta_2(z_k,y_{k+1})). \end{cases}$$

with $\lambda_i = \lambda(z_{k+1})$ and $z_k \subseteq Z$ is the state estimate of $D_{\widetilde{G}}$ at time k.

The diagnoser state transition is given by

$$\begin{cases} (z_{k+1},\lambda(z_{k+1})) = \zeta(z_k,y_{k+1}),\\ \lambda(z_{k+1}) = \zeta_2(z_k,y_{k+1}),\\ z_{k+1} = \zeta_1(z_k,y_{k+1}),\\ \qquad = X_{succ}(z_k,\overset{all}{\rightarrow})\cap\zeta_1(z_k,y_{k+1}),\end{cases}$$

Figure 5 shows the standard diagnoser for the discrete event system model of Figure 4, with $z_0 = \{x_0\}$. Each state of the diagnoser $D_{\widetilde{G}}$, shown as a rounded box in Figure 5, is a set of states of the system. An output symbol and a failure condition are associated with each diagnoser state. For instance, to see the importance of having a complete model for the diagnoser, we suppose at time k the output sequence $''\theta\alpha\mu\xi\eta''$ is observed, then the state estimate is $z_{10} = \{x_{11},x_{12}\}$ and systems condition from z_0 is $\lambda(z_{10}) = N$. The successors of state estimate z_{10} is: $Z_{succ}(z_{10}) = z_{11} = \{x_{13}\}$ or $Z_{succ}(z_{10}) = z_0 = \{x_0\}$. If the next output symbol y_{k+1} is anything other than ξ or θ, we get

$$Z_{succ}(z_{10}) = X_{succ}(z_1,\overset{all}{\rightarrow})\cap\zeta_1(z_1,y_{k+1}) = \emptyset,$$

that means the observation generated after y_k is inconsistent with the model dynamic and the diagnoser cannot proceed. When the output sequence is inconsistent with the model of the system, then we have to revise the model of $\widetilde{G}$ by adding new state(s) and/or new transition(s) respectively in X and Σ, that we believe are missing in the nominal model. This situation may be interpreted as a normal or abnormal situation, because we add new states and/or transitions. Detecting and adding new states and/or transitions in X and/or in Σ of $\widetilde{G}$ is called learning diagnoser. A algorithm of a learning diagnoser is presented in the next section.

5. A ALGORITHM OF A LEARNING DIAGNOSER

A learning diagnoser is a standard diagnosis that tolerant of missing information, *i.e.,* transitions and states, about the system to be diagnosed. The learning diagnoser must be able to learn the true model of the system $\widetilde{G}$, when missing information about the system are presented.

Let a_{new} be a new event detected and not found in Σ of system $\widetilde{G}$, then the new set of input events of $\widetilde{G}$ is given by

$$\Sigma_{new} = \Sigma\cup\{a_{new}\}.$$

A transition $x_d \xrightarrow{a_{new}} x_a$ is ordered pair of state denoting a transition from the state x_d to the state x_a. Let φ' be the extend function transition of φ of the system $\widetilde{G}$ such that

$$\varphi_{new}(x_d,a_i) = \begin{cases} x_a \ \text{ if } \ a_i = a_{new} \ \ \& \ \begin{cases} \Sigma \leftarrow a_{new},\\ \text{and}\\ X \leftarrow x_a \ \text{ if } \ x_a \notin X,\end{cases}\\ \varphi_1(x_d,a_i) \ \text{ otherwise},\end{cases}$$

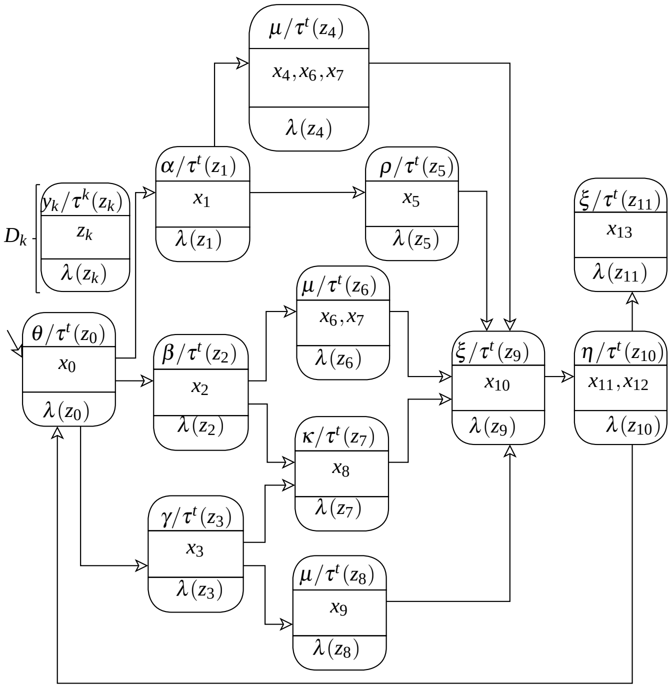

Figure 3. Diagnoser of fuzzy discrete event system model shown in Figure 4, $\lambda(z_i)_{i=0\,to\,10} = N$ and $\lambda(z_{11}) = F_1$.

Let be a dynamic model $\widetilde{G}'$ of $\widetilde{G}$ defines as

$$\widetilde{G}' = extend(\widetilde{G},X',\Pi) = (X\cup X',\Sigma\cup\Pi,Y,\varphi_{new},\widetilde{x}_0).$$

And $\widetilde{G}'$ is called the extension of $\widetilde{G}$ by X' and Π, with X' is the set containing all new states and Π is the set containing all new transitions founded. The set transition Π is empty, if the model G of the system is consistent with the output sequence.

The algorithm presented in Algorithm 1 is the algorithm for the learning diagnoser and evolving fuzzy state automaton.

6. APPLICATION EXAMPLE

Nowadays, the crisis management is an important challenge for medical service and research, to develop new technical of decision support system to guide the decision makers. The crisis management is a special type of collaboration, therefore several aspects must be considered. The more important aspect in a crisis management is the coordination (and communication) between different actors and groups involved in the crisis management. Hence, the capacity to take fast and efficient decisions is a very important challenge for a better exit of crisis. Because the context and characteristics of crisis such as extent of actors and roles, the management becomes more difficult in order to take decisions, but also to exchange information or to coordinate different groups involved. The difficult to take a decision can be also due to random factors, such as stress, emotional impact, road conditions, weather conditions, etc. During the crisis management, it is hard to

```
initialization;
while input is $a_k$ and active state time $t-1$ is $x_i$ do
    read symbol $a_k$;
    $x_j = \varphi_1(x_i, a_k)$;
    $y_j = y_{k+1} = \varphi_2(x_i, a_k)$;
    $X_{succ}(x_i, a_k) = \{\forall\, x_s \in X \mid x_s \in \varphi_1(x_i, a_k)\}$ ;
    if $x_i$ is the start state and time is $t_0$ then
        $X_{pred}(x_j, t) = \emptyset$;
    else
        $X_{pred}(x_j, t) = \big\{\forall\, x_i \in X \mid x_j \in \varphi_1(x_i, a_k)\big\}$;
    end
    if ($X_{succ}(x_i, a_k) \cap \zeta_1(z_k, y_{k+1}) \neq \emptyset$) then
        if ($|X_{pred}(x_j, t)| = \emptyset$) then
            $X_{act} = \tilde{x}_0$;
            $X_{succ}(x_0, a_k) = \big\{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\big\}$ ;
        else if ($|X_{pred}(x_j, t)| = 1$) then
            single mv of all active states ;
            $\mu^t(x)$ of each state $x \in X_{act}$ is computed by ;
            $\mu^t(x_j) = \Psi_1(\mu^t(x_j), \delta(x_i, a_k, x_j))$ ;
            $X_{act} = \big\{(x_j, \mu^t(x_j)\big\}$ ;
            $X_{succ}(x_j, a_k) = \big\{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\big\}$ ;
        else
            active state have been forced to take different
            several mv ;
            $m = |X_{pred}(x_j, t)|$ ;
            for $i = 1$ to $m$ do
                $\mu_i = \Psi_1(\mu^{t-1}(x_i), \delta(x_i, a_k, x_j))$;
            end
            $\mu^t(x_j) = Max(\mu_1, \mu_2, \cdots, \mu_{m-1}, \mu_m)$;
            $X_{act} = \big\{(x_j, \mu^t(x_j)\big\}$ ;
            $X_{succ}(x_j, a_k) = \big\{\forall\, x_s \in X \mid x_s \in \varphi_1(x_j, a_k)\big\}$ ;
        end
        Diagnoser method ;
        go to $D_k$;
    else
        go to inconsistency;
        detection of new transition and/or state;
        $X_{succ}(x_i, a_k) \cap \zeta_1(z_k, y_{k+1}) = \emptyset$;
        we suppose for all new transition;
        $\delta(x_i, a_k, x_j) = 0$;
        if ($x_j \in X \,\&\, a_k \in \Sigma$) then
            new transition between $x_i$(past state) to $x_j$
            (active state) ;
        else if $x_j \in X$ & $a_k \notin \Sigma$ then
            update $\Sigma$;
            $\Sigma \leftarrow a_k$ ;
        else
            update $X$ and $\Sigma$;
            $X \leftarrow x_j$ ;
            $\Sigma \leftarrow a_k$ ;
        end
    end
end
```

Algorithm 1: Evolving fuzzy finite state automaton

say exactly an actor's stress has changed from low to high. For this reason, it is important to integrate these factors in the model of crisis management for decision-making. The *FFA* presented above is used to takes into account the stress of the actors involved in the crisis management.

6.1. Our *FFA* model of crisis management

In this paper, we propose a model (no generic model) applied on the team *SAMU* [1] from Hospital of Troyes in France, during *TEAN* [2] exercise.

The team of *SAMU* is composed of the following actors:

- Rear Base [3] (*RB*): Operations Coordination,
- Communication Center (*CC*): collecting information and sharing with *RB*,
- First Team: first intervention, sending the first evaluation (result) about the crisis to the *CC*,
- Advanced Medical Post (*AMP*): Intervention and evacuation of victims, sending the complete evaluation to the *CC*.

The *FSA* of the *TEAN* exercise is shown in Figure 4.

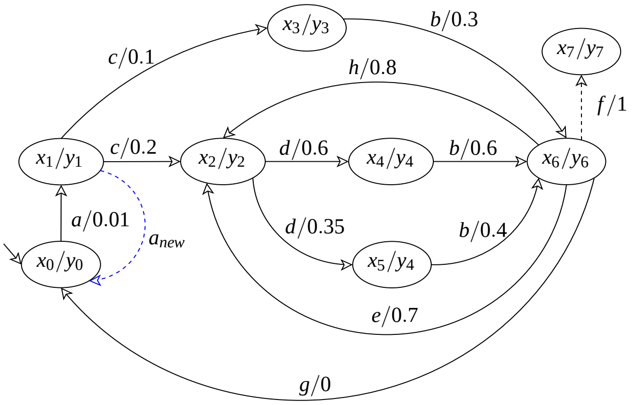

Figure 4. A example of modelisation of a scenario of crisis with finite state automaton and the weight corresponds to the stress of actors involved.

The discrete event model showed in Figure 4 for *TEAN* exercise, allows one hand to monitor the communication and coordination between various groups involved in crisis management, and also to supervise some specific behaviors that are critical situations. Thus the factor's stress of the actors involved is estimated for decision-making.

Consider the *FFA* in Figure 4 with several transition overlaps and several output labels. It is specified as:

$$\widetilde{G}_n = (X, \Sigma, \delta, Y, \widetilde{x}_0, F),$$

[1] SAMU is Service Emergency Medical Assistance.
[2] TEAN is the name of the exercise.
[3] Other word, Rear Base is decision makers

The dashed line in Figure 4, between states 6 and 7 represents a critical event. The occurrence of event $''f''$ bring the system in or critical mode corresponding to state x_7 and $\omega_{i,j}$ is the stress of actors involved in crisis management.

In this example

$X = \{x_0, x_1, \cdots, x_7\}$, is the set of states, which occur with different, membership degree $(\mu^t(x_0), \cdots, \mu^t(x_7))$.

$\Sigma = \{a, b, c, d, e, f, g, h\}$, set of input symbols,

$Y = \{y_1, y_2, y_3, y_4, y_6, y_7\}$, set of output events,

$\widetilde{x}_0 = \big\{(x_0, \mu^{t_0}(x_0) = 0)\big\}$, starting state,

$$\lambda(x_i) = \begin{cases} F_1(\text{abnormal mode}), & \text{if } \quad \text{i=7}, \\ N(\text{normal mode}), & \text{otherwise}. \end{cases}$$

Table 1. List and definition of the states.

States	Definition
x_0	No crisis
x_1	Onset Crisis
x_2	Information received at the communication center (CC)
x_3	Information arrived at the police center
x_4	Information received at the Emergency department
x_5	Information arrived at the Advanced Medical Post (AMP)
x_6	Information received at the accident area
x_7	The model is unpredictable for this crisis situation

Table 2. List and definition of outputs.

Output labels	Definition
y_0	No coming call
y_1	Accident is happen
y_2	Information arrived to CC
y_3	Information arrived to police office
y_4	Preparation of the Intervention Team
y_5	Preparation of the AMP
y_6	New Actors arrived in the accident area
y_7	uncontrolled situations (conditions)

Table 3. List and definition of the transitions (events).

events	Definition
a	A call from (or about) a accident
b	Sending Team to the accident site
c	Sending information to CC and police office
d	Sending information to Emergency
e	Sending the first evaluation to CC
h	Sending final evaluation to CC
f	End of crisis management without success
g	End of crisis management with success

In this example, we suppose at the beginning $\mu^{t_0}(x_0) = 0$ (i.e, stress level is very low) and all the other *mv* are computed by using approaches presented in section 3.

Assuming that $\widetilde{G}_n$ starts operating at time t_0 and the next three input are ”**a**” respectively (one at a time), active states and their $mv's$ at each time step are as follows.

- **at time t_0**
  $X_{act}(t_0) = \{(x_0, \mu^{t_0}(x_0)\}$

  $$\begin{cases} X_{succ}(x_0, a) = x_1, \\ X_{succ}(x_0, \overset{all}{\rightarrow}) = \{x_1\}. \end{cases}$$

  $X_{succ}(x_j, \overset{all}{\rightarrow})$ is the set of all successors of state x_j,

- **at time t_1, input is ”a”**
  $X_{act}(t_1) = \{(x_1, \mu^{t_1}(x_1))\}$,
  $Y_{act}(t_1) = \{(y_1, \tau^{t_1}(z_1))\}$, and $\tau^{t_1}(z_1) = \tau^{t_1}(x_1) = \mu^{t_1}(x_1)$ at time t_1 the weight corresponding to the stress of the people involved is $\omega_{0,1} = 0.01$ and this weight is estimated by the expert of the crisis management.
  $X_{pred}(x_1, t_1) = x_0$, and $|X_{pred}(x_1, t_1)|$ is the number of predecessors of active state x_1. $|X_{pred}(x_j, t)| = 1$, then, the active state x_1 is not forced to take multi-membership. $X_{succ}(x_1, c) = \{x_2, x_3\}$,

6.2. Diagnoser model of $TEAN$ exercise

The standard diagnoser for the fuzzy discrete event system of crisis management model illustrated in Figure 4 is shown in Figure 5, with $z_0 = \{x_0\}$. Each state of the diagnoser $D_{\widetilde{G}_n}$, shown as a rounded box in Figure 5, is a set of states of the system. An output symbol corresponding to the operating condition of the system is associated with each diagnoser state. For example, to see the importance of having a complete model for the diagnoser, we suppose at time t_1 the output sequence $''y_0 y_1''$ (*see* Figure 4) is observed, then the state estimate is $z_1 = \{x_1\}$ and the operating condition from z_0 is $\lambda(z_1) = N$. The successors of state estimate z_1 is: $Z_{succ}(z_1) = \{z_2, z_3\} = \{x_2, x_3\}$. If the next output symbol y_{t+1} is y_0, we get

$$Z_{succ}(z_1) = X_{succ}(z_1, \overset{all}{\rightarrow}) \cap \zeta_1(z_1, y_{t+1}) = \emptyset,$$

that means the observation generated after y_1 is inconsistent with the model dynamic and the diagnoser cannot proceed. When the output sequence is inconsistent with the system's model, then we have to revise the model of $\widetilde{G}_n$ by adding in this application a new transition (a_{new}) from the state x_1 to the state x_0(s) (see Figure 4). This situation may be interpreted as a normal or abnormal situation. Detecting and adding new states and/or transitions in X and/or in Σ of $\widetilde{G}$ is called learning diagnoser.

7. CONCLUSION

In this paper, we have dealt with the failure diagnosis of fuzzy finite state automaton for systems operating in non-stationary environment. We have presented in our paper, the definition

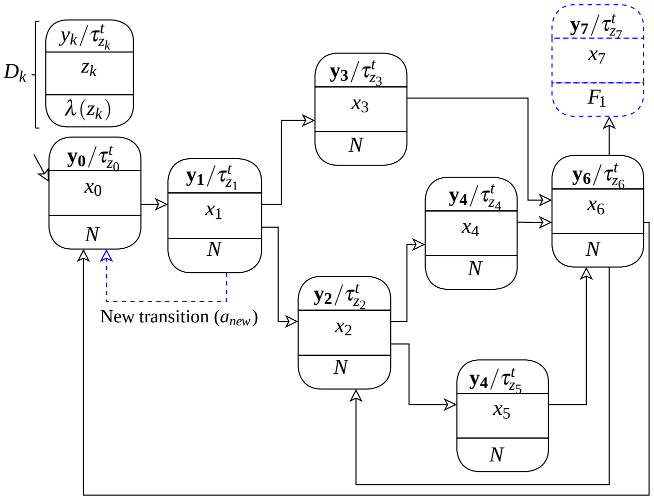
D_k
$y_k/\tau^t_{z_k}$
z_k
$\lambda(z_k)$
$\mathbf{y_0}/\tau^t_{z_0}$
x_0
N
$\mathbf{y_1}/\tau^t_{z_1}$
x_1
N
$\mathbf{y_3}/\tau^t_{z_3}$
x_3
N
$\mathbf{y_7}/\tau^t_{z_7}$
x_7
F_1
$\mathbf{y_4}/\tau^t_{z_4}$
x_4
N
$\mathbf{y_6}/\tau^t_{z_6}$
x_6
N
New transition (a_{new})
$\mathbf{y_2}/\tau^t_{z_2}$
x_2
N
$\mathbf{y_4}/\tau^t_{z_5}$
x_5
N

Figure 5. Diagnoser of fuzzy discrete event system model shown in Figure 4.

of a crisp discrete event system and fuzzy discrete event system. The main advantage of fuzzy finite state automaton, to handle imprecise and uncertain data is presented. We have formalized the construction of the learning diagnoser based on evolving fuzzy finite state automaton that are used to perform fuzzy diagnosis. In particular, we have propose a algorithm for learning diagnoser based on evolving fuzzy finite state automaton that allows to add new transitions and states. The newly proposed diagnoser approach allows us to deal with the problem of failure diagnosis for fuzzy discrete event system, which many better deal with the problem of fuzziness, impreciseness and uncertainness in the failure diagnosis.

The potential application of learning diagnosis based on fuzzy finite state automaton is in solving the ambiguity in a fault diagnosis problem especially in the case of multiple faults.

Future work will focus on the proposal of fuzzy states of crisis management by using fuzzy finite automaton that takes into account of a random vector as such the stress, weather condition and emotional impact of the actors involved in crisis management.

ACKNOWLEDGMENT

This work is supported by the CPER prject AidCrisis, sponsored by the Champagne-Ardenne region and the French ministry of higher education and research.

REFERENCES

Cabasino, M. P., & Alessandro Giua, C. S. (2010). Fault detection for discrete event systems using petri nets with unobservable transition. *Automatica, vol. 46*, pp. 1531-1539.

Cao, Y., & Ying, M. (2005). Supervisory control of fuzzy discrete event systems. *IEEE transactions on systems, Man, and Cybernetics-part B: Cybernetics, vol. 35 (2)*, pp. 366-370.

Cao, Y., & Ying, M. (2006). Observability and decentralized control of fuzzy discrete-event systems. *IEEE Trans. Fuzzy Syst., vol. 14*, pp. 202-216.

Cassandras, C. G., & Lafortune, S. (1999). Introduction to discrete event systems. *Boston, MA:Kluwer*.

Doostfatemeh, M., & Kremer, S. C. (2004). The significance of output mapping in fuzzy automato. *Proceedings of the 21th Iranian Conference on Electrical Engineering (ICEE)*.

Doostfatemeh, M., & Kremer, S. C. (2005). New directions in fuzzy automaton. *International Journal of Approximate Reasoning, vol. 38*, pp. 175-214.

Dzelme-Berzina, I. (2009). Mathematical logic and quantum finite state automata. *Theoretical Computer Science, vol. 410*, pp. 1952-1959.

Gerasimos, G. R. (2009). Fault detection and isolation based on fuzzy automaton. *Information Sciences, vol. 179*, pp. 1893-1902.

Kwong, R. H., & Yonge-Mallo, D. L. (2011). Fault diagnosis in discrete-event systems: Incomplete models and learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 41 (1)*, pp. 118 - 130.

Liu, F., & Qiu, D. (2009a). Diagnosability of fuzzy discrete-event system: A fuzzy approach. *IEE Transactions on Fuzzy Systems, vol. 17 (2)*, pp. 372-384.

Liu, F., & Qiu, D. (2009b). Diagnosability of fuzzy discrete event systems: A fuzzy approach. *IEEE Transactions on fuzzy system, vol. 17 (2)*, pp. 372-384.

Luo, M., Li, Y., Sun, F., & Liu, H. (2012). A new algorithm for testing diagnosability of fuzzy discrete event systems. *Information Sciences, vol. 185*, pp. 100 - 113.

Moamar, S.-M., & Billaudel, P. (2012). Abrupt and drift-like fault diagnosis of concurent discrete event systems. *Machine Learning and Applications (ICMLA), vol. 2*, pp. 434-439.

Moghari, S., Zahedi, M. M., & Ameri, R. (2011). New direction in fuzzy tree automata. *Iranian Journal of Fuzzy Systems, vol. 8 (5)*, pp. 59-68.

Mukherjee, K., & Ray, A. (2014). Statesplittingandmerginginprobabilisticfinitestate automataforsignalrepresentationandanalysis. *Signal Processing, vol. 104*, pp. 105-119.

Patela, A. M., & Joshi, A. Y. (2013). Modeling and analysis of a manufacturing system with deadlocks to generate the reachability tree using petri net system. *International Conference On DESIGN AND MANUFACTURING, IConDM 2013, vol. 64*, pp. 775-784.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transaction On Automatic Contol*, *vol. 40 (9)*, pp. 1555-1575.

Sardouk, A., Mansouri, M., Merghem-Boulahia, L., & Gaiti, D. (2013). Crisis management using mas-based wireless sensor networks. *Computer Networks*, *vol. 57*, pp. 29-45.

Thomas, W. (1990). Handbook of theoretical computer science. *Elsevier, B.*

Traore, M., Moamar, S.-M., & Billaudel, P. (2013). Learning diagnoser and supervision pattern in discrete event system : Application to crisis management. *Annual Conference of the Prognostics and Health Management Society, ISBN-978-1-936263-06-6, New Orleans, USA*, 694-701.

BIOGRAPHIES

Dr. Moussa Traore is a Postdoctoral fellow at the University of Technology of Troyes, France working under supervision of prof. Eric Chatelet. He is working on Fault Diagnosis in discrete event systems and fuzzy discrete event systems. He is working on an AidCrisis project sponsored by Champagne-Ardenne region and the French ministry of higher education and research. He obtained his *Ph.D* degree (Control System and Signal Processing) from University Lille 1 in France. He received his Master of Science (Optimization and safety of functioning Systems) from University of Technology of Troyes, France, 2006 and Bachelor's degree in Electrical Engineering from Faculty of Sciences and Technology of Nouakchott, Mauritania, 2004. His research area is diagnostic and prognostic of dynamic systems in nonstationary environment, for continuous and discrete systems. His research work allowed him to publish 4 journal papers, 7 papers with proceedings at international conferences and 3 papers with proceedings at national (France) and international conferences (in French) in the area of diagnosis, prognosis and predictive maintenance. He served as International reviewer for several International Conferences.

Dr. Hossam A. Gabbar Dr. Hossam A. Gabbar is Associate Professor in the Faculty of Energy Systems and Nuclear Science, and cross appointed in the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology (UOIT). He obtained his Ph.D. degree (Safety Engineering) from Okayama University (Japan), while his undergrad degree (B.Sc.) is in the area of automatic control from Alexandria University, Egypt. He is specialized in safety and control engineering where he worked in process control and safety in research and industrial projects in Japan and Canada. Since 2004, he was tenured Associate Professor in the Division of Industrial Innovation Sciences at Okayama University, Japan. And from 2001, he joined Tokyo Institute of Technology and Japan Chemical Innovative Institute (JCII), where he participated in national projects related to advanced distributed control and safety design and operation synthesis for green energy and production systems. He developed new methods for automated control recipe synthesis and verification, safety design, and quantitative and qualitative fault simulation.

He is a Senior Member of IEEE, the founder of SMC Chapter - Hiroshima Section, the founder and chair of the technical committee on Intelligent Green Production Systems (IGPS), and Editor-in-chief of International Journal of Process Systems Engineering (IJPSE), president of RAMS Society, and editorial board of the technical committee on System of Systems and Soft Computing (IEEE SMCS). He is invited speaker in several Universities and international events, and PC / chair / co-chair of several international conferences. Dr. Gabbar is the author of more than 110 publications, including books, book chapters, patent, and papers in the area of safety and control engineering for green energy and production systems. His recent work is in the area of risk-based safety and control design for energy conservation and supply management, and smart grid modeling and planning with distributed generation.